

## useId Hook:

Connect with Me:

LinkedIn : <https://www.linkedin.com/in/priya-bagde>

GitHub : <https://github.com/priya42bagde>

LeetCode : <https://leetcode.com/priya42bagde/>

YouTube Channel : [https://youtube.com/channel/UCK1\\_Op30\\_pZ1zBs9l3HNyBw](https://youtube.com/channel/UCK1_Op30_pZ1zBs9l3HNyBw)  
(Priya Frontend Vlogz)

- `useId()` is one of the latest hooks added to **React 18**.
- The `useId()` hook provides a way to **generate unique IDs that persist between re-renders**.
- It ensures that generated IDs i.e, **returns a string** are **unique across the entire React app**, and **until the component that uses the ID is removed from the DOM**. Once the component is **re-insterted**, the **generated ID will be different**.
- The **main purpose** of the `useId()` hook is to generate unique IDs **for HTML form elements**. It simplifies the process of generating unique IDs **when creating form inputs and labels in React**.
- The `useId()` hook **doesn't take any parameters**.
- **Don't use `useId` hook to generate keys. It's main purpose is to generate unique ids for Form element's id attribute.**

**What if we re-use this Form component for multiple times? Will both components have same id: No.** Each instance of the component will have different and unique id across the React app.

**What if we want multiple unique id in the same component?** We can create as many unique id as we want but wait you may not need it. Suppose we require not 3 but 10 or 15 ids. With this manner we will end up creating 10 to 15 unique ids by calling hooks for 10 to 15 times. Seems odd right? Usually in such cases, we can prefix single generated id and reuse it at multiple places.

```
function Form() {  
  const idForFirstName = useId();  
  const idForLastName = useId();  
  const idForEmail = useId();  
  
  return (  
    <>  
      <label htmlFor={idForFirstName}>First Name</label>  
      <input id={idForFirstName} type="text" placeholder="First Name" />  
  
      <label htmlFor={idForLastName}>Last Name</label>  
      <input id={idForLastName} type="text" placeholder="Last Name" />  
  
      <label htmlFor={idForEmail}>Email</label>  
      <input id={idForEmail} type="email" placeholder="Email" />  
    </>  
  )  
}
```

```
function Form() {  
  const id = useId();  
  
  return (  
    <>  
      <label htmlFor={` ${id} --firstName`} >First Name</label>  
      <input id={` ${id} --firstName`} type="text" placeholder="First Name" />  
  
      <label htmlFor={` ${id} --lastName`} >Last Name</label>  
      <input id={` ${id} --lastName`} type="text" placeholder="Last Name" />  
  
      <label htmlFor={` ${id} --email`} >Email</label>  
      <input id={` ${id} --email`} type="email" placeholder="Email" />  
    </>  
  )  
}
```